

Appendix

```

ON PACKET P ARRIVING  $queue_i$  OF  $source_i$ 
1  enqueue(p,  $queue_i$ );
2  /*  $A$  is set of active traffic sources */
3  if  $i \notin \hat{A}$ 
4       $\hat{A} = \hat{A} - \{i\}$ ;
5       $start\_tag_i = virtual\_time$ ;
6       $finish\_tag_i = start\_tag_i$ 
7          +  $\sum_{i \in A} (r_i) / r_i + size(p) / speed_i$ ;
8       $credit_i = 0$ ;

EXTRACTING PACKET FOR TRANSMITTING
1   $l = \min_{start\_tag_k} (k \in A)$ ;
2  if  $credit_l \leq 0$  /* handling non-leading flows */
3      if (l can send)
4          PKT_SEND( $l, l$ )
5      else
6          /* l cannot send due to channel error */
7          /* find flow with smallest start_tag to substitute i */
8           $g = \min_{start\_tag_k} (k \in \hat{A}, k \text{ can send})$ 
9          if (g exists)
10             /* transmit for g, update tags, credit for both */
11             PKT_SEND( $g, l$ );
12             else /* All sources are in channel error */
13                 IDLE;
14     else /* handling leading flows */
15         /* decide whether let leading flow transmit or compensate */
16          $flag = compensation\_flag(l)$ ; /* 1: compensate, 0: normal */
17          $g = \min_{comp\_tag_k} (k | k \in A, credit_k < 0, k \text{ can send})$ ;
18         if ( $l \text{ can send}$ ) and ( $(!flag)$  or ( $flag$ ) and ( $g$  !exists))
19             PKT_SEND( $l, l$ );
20         else if (g exists)
21             PKT_SEND( $g, l$ );
22         else /* All sources are in channel error */
23             IDLE;
24         /* check if some flow becomes inactive */
25         /* adjust other flows in  $A$  accordingly */
26         HANDLE_IDLE_FLOW;

PKT_SEND( $g, l$ )
/* send packet from  $source_i$ , update tags, credits for both */
1   $p_l = dequeue(l)$ ;
2  /* update start and finish tag only for  $l$  */
3   $start\_tag_l = \max(virtual\_time, finish\_tag_l)$ ;
4   $finish\_tag_l = start\_tag_l + \sum_{i \in A} (r_i) / r_l + size(p_g) / speed_l$ ;
5  if ( $l \neq g$ )
6       $credit_l += size(p_g) / speed_l$ ;
7       $credit_g -= size(p_g) / speed_l$ ;
8       $comp\_tag_l = -size(p_g) / (speed_l + credit_l)$ ;
9       $comp\_tag_g = -size(p_g) / (speed_g + credit_g)$ ;

HANDLE_IDLE_FLOW( $j, i$ )
/* adjust flows credit counters once some flow becomes inactive */
1   $\hat{A} = \hat{A} - \{i\}$ 
2  if  $\exists i, s.t. empty(queue_i)$ 
3      for ( $k \in \hat{A}$ )
4           $credit_k = credit_k + credit_i * r_i / \sum_{k \in \hat{A}} (r_k)$ ;

```